

## What's New About *TCP Linda*® ?

There's a new Linda now in 2003, *TCP Linda*, from Scientific Computing Associates, experts in parallel and distributed computing since 1980. *Linda* is a unique programming tool that enables developers to develop parallel versions of existing applications efficiently and quickly to run on standard distributed memory multiprocessors and clusters or networks of computers. A major use of *TCP Linda* is to provide high performance for parallel *Gaussian '03*®, the latest release of the dominant quantum chemistry code from Gaussian, Inc.

The original *Linda*, introduced in the mid-1980s, was the first commercial product to implement a virtual shared memory (VSM), now popularly known as tuplespace technology for supercomputers and large workstation clusters. It is used at hundreds of sites worldwide. *TCP Linda*'s semantics are unchanged, that is, its tuplespace remains content-addressable, not address-based, which makes it easier to build applications and fully utilize hardware capacity. Cost-effectiveness, speed, and ease of use are just some of the advantages *TCP Linda*. Other advances include:

- Simpler to use
- Streamlined runtime system
- Significantly higher transfer rates for large tuples
- Better memory management
- Optimized for large data fields
- More modular architecture

Previous *Linda* releases were based on UDP. The new release is based on TCP. Over the last decade, as TCP has emerged as the dominant protocol for networking, developers of networking hardware and software have tended to focus on optimizing its performance. Our move to TCP leverages this optimization work. In particular, compared to previous releases of *Linda*, *TCP Linda* delivers significantly higher transfer rates for large tuples, and competitive transfer rates even for very small tuples.

TCP is inherently simpler to use than UDP so we have been able to restructure and streamline our runtime system. As a result, we devote less time to communication "plumbing" and more to optimizations. *TCP Linda* has benefited from this effort, particularly in the area of memory management, where we have seen two major improvements: better use of dynamic memory and effective optimization for large data fields.

The UDP release dynamically allocates memory within the runtime system. This necessitates imposing on the user a malloc facility that can be used concurrently by the user's code and the runtime system. The malloc facility used in the UDP release is robust, but has begun to show its age. In addition, the UDP malloc maintains two storage pools (one for the user, one for the runtime system), which reduces the efficiency with which dynamic memory was recycled. This could be a particular problem with codes that moved large quantities of data through tuplespace, as it could effectively double memory requirements. Now, *TCP Linda* uses a different management scheme in which the

runtime system never dynamically allocates space. As a result, there is no risk of concurrent malloc use, and thus the native malloc can be used. Large dynamic memory requests for the *TCP Linda* runtime system are made in the user context during the invocation of *outs*. Any remaining small dynamic memory requests are handled from a fixed-at-startup (but user configurable) memory pool.

The simplicity of the *TCP Linda* runtime system also enabled an optimization that can be quite effective for large data fields. Normally, every data field of a tuple being “*outed*” is copied to internal storage before the *out* completes. The user is then free to modify variables used in those fields, without risk of altering the contents of tuple space. For large data fields, this means dynamically creating an internal copy. The new optimization allows the user to assert that a given data field will not be modified during the “lifetime” of the tuple. The *TCP Linda* runtime system then need only track the address of the large data field, with the result that no extra space is required and that no cycles are spent copying data. In fact, the mechanism is even more powerful—a single *out*, with appropriate synchronization and in the presence of a sequence of *rds*, could be used to produce, in effect, a set of different tuples.

*TCP Linda* has embraced the trend towards threaded applications, which has several implications. First, it avoids the problems that arose from the UDP release’s use of a signal handler. In particular, user code need not worry about interrupted system calls due to the UDP runtime system’s handling of SIGIO or about difficulties with sharing SIGALARM. Second, it lays the foundation for the development of a thread-safe implementation that would allow user level *Linda* operations in multiple threads (which will be implemented in a future release). However, it does mean that now applications have to be compiled in a “thread aware” manner. While this is well supported on most major platforms, there may be some difficulties with using third-party binary libraries. For this reason, we will continue to maintain the UDP release until it is clear that any difficulties have been addressed.

In combination with the new *TCP Linda* runtime system, we have modified the process that “launches” *Linda* executions. The overall architecture has been made more modular: configuration interpretation and node selection are handled by one module, the launching of the subprocesses by another. The interface between the two is textual, making it straightforward to mix-and-match alternate implementations of both modules. Benefits include: elimination of some (rare) start up races, better compatibility with ssh, and cleaner support for launching “by-hand” (for example, to setup runs under a special debugger).

Worldwide distribution of *TCP Linda* is from Scientific Computing Associates, New Haven, CT.

**To order:** [sales@lindaspaces.com](mailto:sales@lindaspaces.com)

Web: [www.lindaspaces.com](http://www.lindaspaces.com)

Voice: 203-777-7772

Fax: 203-776-4074

Worldwide distribution of *TCP Linda* for parallel Gaussian is available directly from Gaussian, Inc., North Haven, CT

***To order TCP Linda for parallel Gaussian: [sales@gaussian.com](mailto:sales@gaussian.com)***

Web: [www.gaussian.com](http://www.gaussian.com)